

Fig. 1

FIG. 2

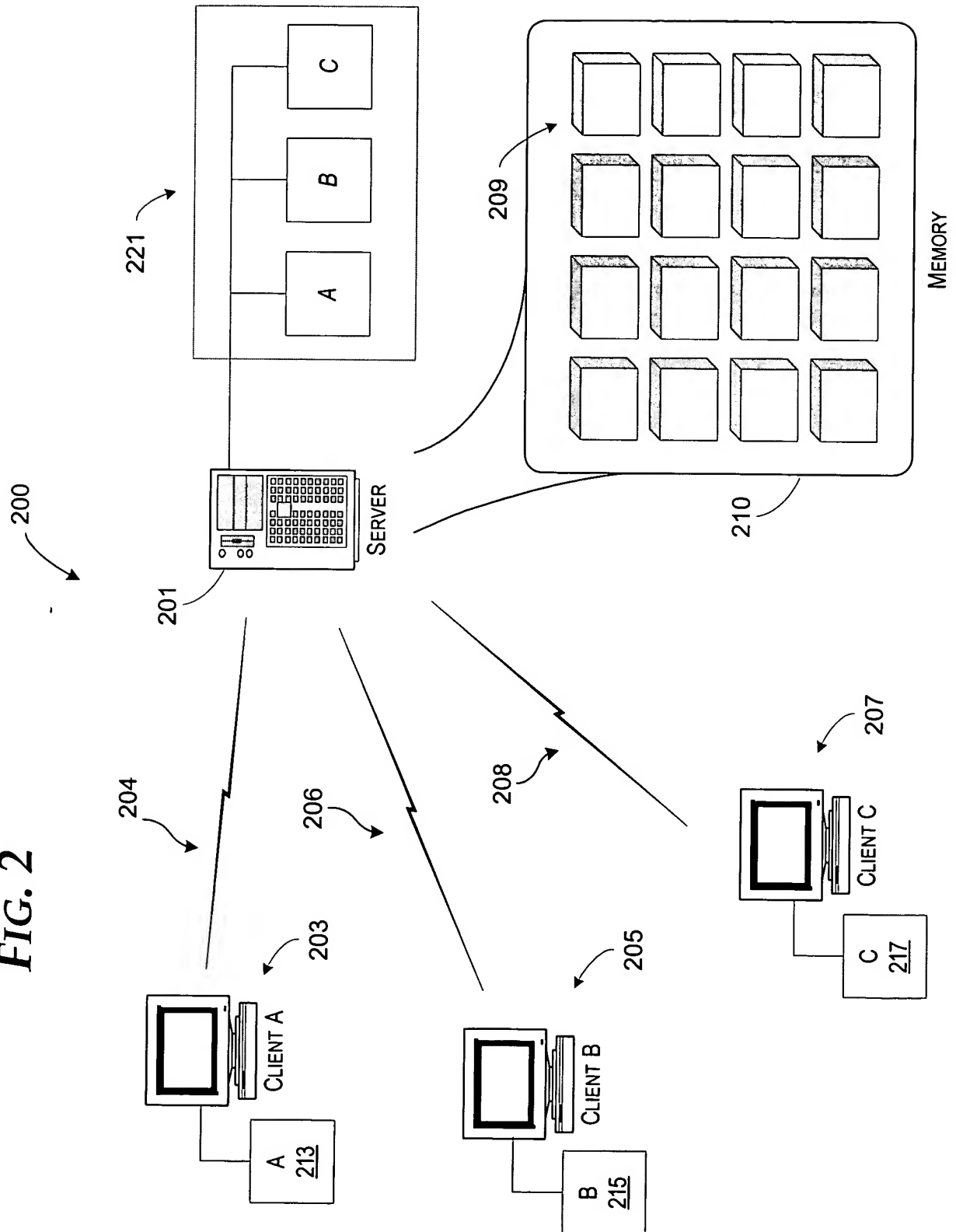


FIG. 3

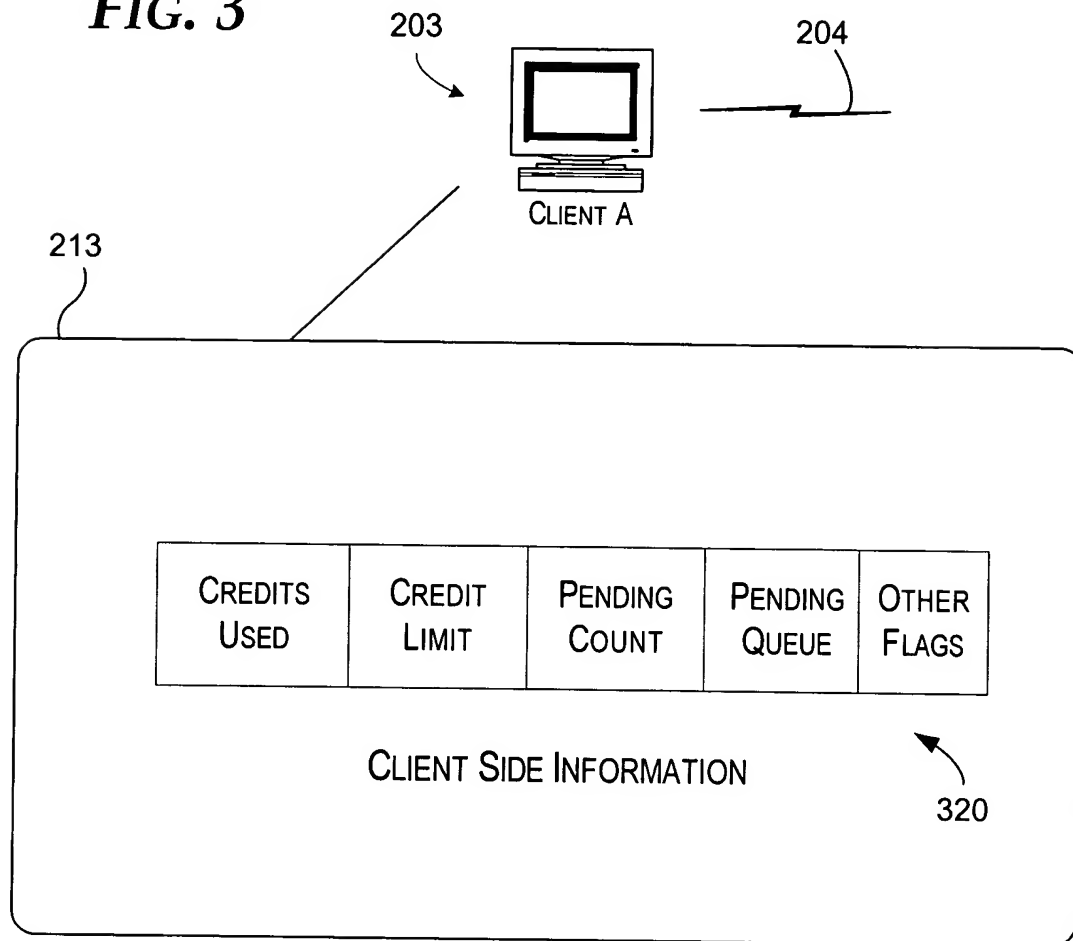
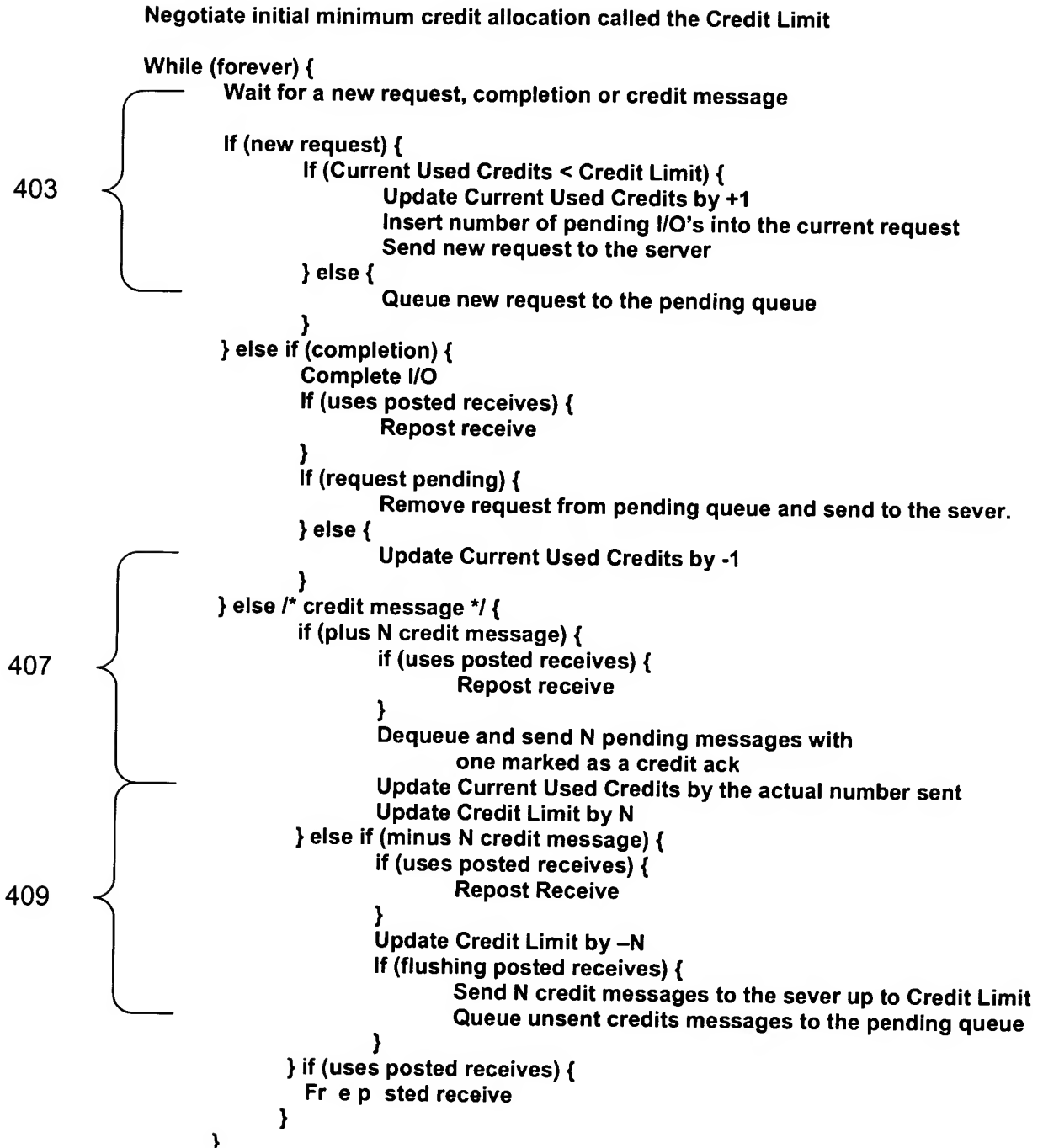


FIG. 4



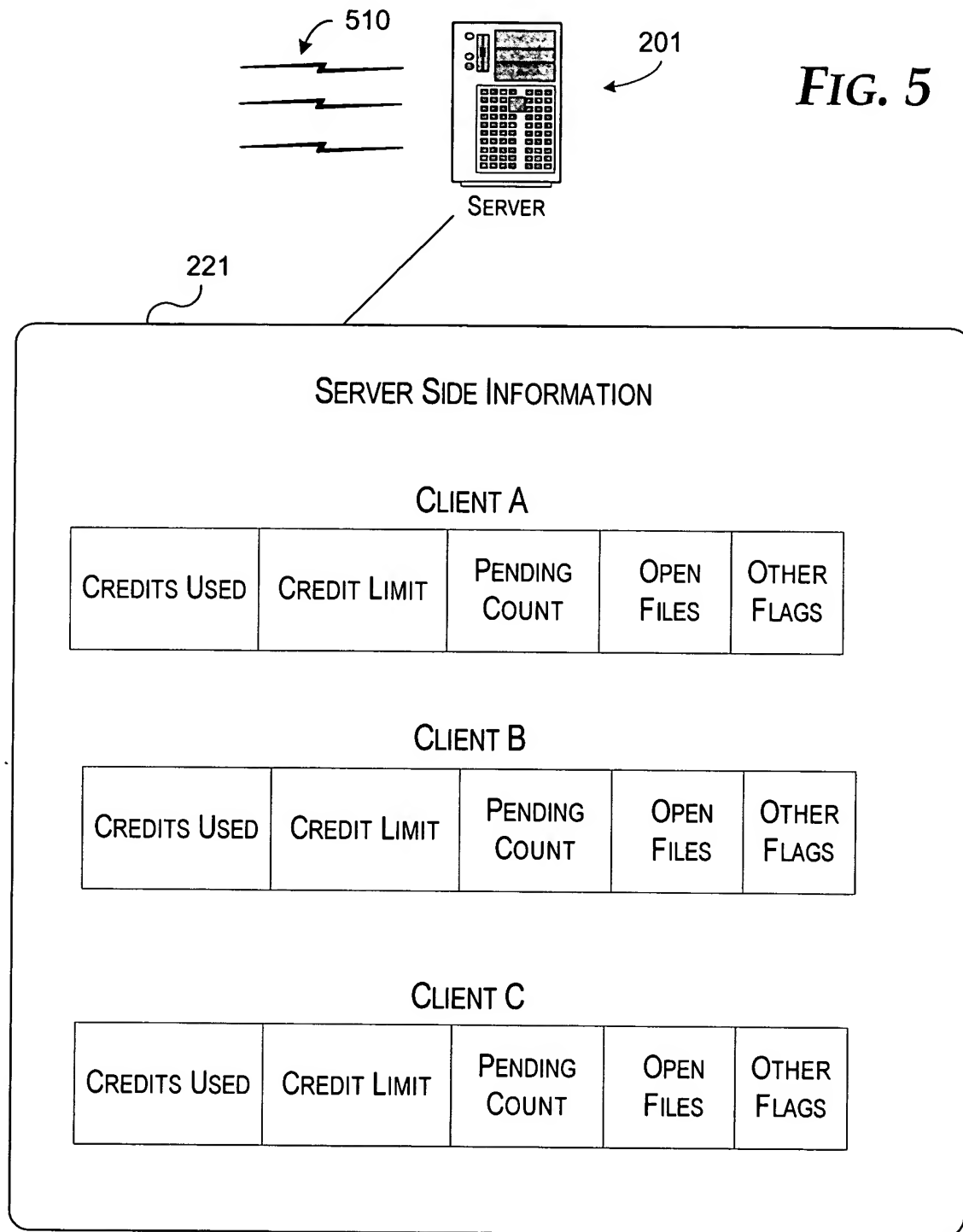


FIG. 6

```

/* i is the index for a particular client instance. */
/* i,j is the index for file j on client instance i */
While (forever) {
    Wait for new client connection or
    new client I/O request or
    credit timer expires or
    credit message

    if (new connection) {
        Process Connection Request
        Negotiate minimum credits for the connection
        Accept connection if credits available otherwise reject
        Set Credit Timer if connection accepted.
    } else if (new client I/O request) {
        /* CP = Client hint on number Pending requests to be sent */
        If (CP of new request > CPi) then {
            Store CP in CPi
            Set Credit Timer if not already set
        }

        Dispatch client I/O request
    } else if (credit message) {
        Acknowledge credit message
        If all credits acknowledge then mark connection as acknowledged
    } else if (credit timer expired) {
        Compute; {
            Total Client Connections (TC) =  $\sum_i 1$ 
            Total Files open (TF) =  $\sum_{i,j} 1$ 
            Total pending requests (TR) =  $\sum_i CP_i$ 

            /* CUi is the number of credits in use on connection i */
            Total credits used (TU) =  $\sum_i CU_i$ 
        }

        for each client connection i {
            if (connection i waiting for ack) {
                do nothing or disconnect if wait time expired
            } else {
                if ((TU + TR) > MC) {
                    if (be fair to files) {
                        NCLi = MC * CF/TF
                    } else /* be fair to connections */ {
                        NCL = MC/TC
                    }
                } else {
                    NCLi = infinite;
                }

                Adjust NCLi to meet minimum negotiated

                if (CLi > NCLi) then {
                    Send minus delta credit message of NCi - NCLi
                } else if (((CLi * Completion Factor) - CUi) > 0) then {
                    Send minus delta credit message of CLi - CUi
                } else if (CPi > 0) {
                    Send positive delta credit message of
                    min(CPi, NCLi - NCi, MC - TU)
                    increase TU by N and reduce TR by N
                }
            }
        }
    }
}

```